CSE696, Spring 2021 **Problem Set 1** **Due Fri. 2/26**

Reading:

Saturday's lecture will be in Chapter 6 of the Arora-Barak draft text, focusing on sections 6.1–6.3 and 6.7. What I really intend to do with 6.7 is make the Cook-Levin proof work for circuits that have "oracle gates" with some number m of input wires whose values define a binary string y, and whose output wire(s) tells whether y belongs to the oracle. Then next week we will be in chapter 7 and sections 9.1 of chapter 9, plus just the first page of section 9.2.

(1) Write $\sum_3^0$ definitions for the following predicates of a single deterministic Turing machine M_i:

(a) There is a string u such that M_i accepts all strings that begin with u.

(b) $L(M_i)$ is co-finite—that is, the complement of $L(M_i)$ is finite.

(c) $L(M_i)$ can be decided in $O(n^2)$ time.

Note that you can't assume that M_i itself is total. You do not need to show that $\sum_3^0$ is optimal. (18 pts. total)

(2) Recall that a Turing machine M_i runs in time $t(n)$ if for all x, $M_i(x)$ halts within $t(|x|)$ steps. Define:

1. $L_1 = \{i : M_i \text{ runs in time } n^2 + 2\}$.
2. $L_2 = \{i : \text{for some } k,\ M_i \text{ runs in time } n^2 + k\}$.
3. $L_3 = \{i : \text{for some } k,\ M_i \text{ runs in time } kn^2 + k\}$.

Note that unlike the predicates in problem (1), these are not index sets but rather properties of individual machines.

(a) Show that L_1 is $\prod_1^0$-complete; that is, complete for co-RE under many-one reductions. (15 pts.)

(b) Show that L_3 is complete for $\sum_2^0$ under many-one reductions. Does your proof also work for L_2? (18 + 3 = 36 pts. total)

(3) (a) Write a $\sum_3^0$ definition for the predicate $K \leq_m^p L(M_i)$, that is, for the index set of the languages that are complete for RE under polynomial-time many-one reductions. Represent the language K by a fixed non-total TM M_k such that $L(M_k) = K$ and use a recursive presentation $[P_j]$ of polynomial-time bounded machines computing the functions in FP. (The lecture notes on equivalence of the languages of non-total machines come into play here. 12 pts.)

(b) Then write a $\sum_4^0$ definition of the predicate $K \leq_T^p L(M_i)$, that is, for the class of languages that are polynomial-time Turing complete for RE. (This will resemble innards of the proof of the "weak hierarchy theorem" in lecture notes. If one removes the words "polynomial time" here, then it is known that $\sum_4^0$ cannot be improved—this is an exception to the third level being the limit for "nice" and "natural" computability concepts. But having those words means you can now take $[P_j]$ to be a fixed recursive presentation of polynomial-time bounded oracle Turing machines, and maybe that allows saving an alternation. Hmmm...18 pts. for the definition and maybe more for fruitful further riffs. I don't know how to prove $\sum_4^0$-completensss, anyhow.)

(4) Suppose B_2 polynomial-time Turing reduces to B_1, that is, to SAT. Show that then the polynomial hierarchy "collapses." Say the smallest class $\mathcal{C}$ that it collapses to, i.e., so that $\mathsf{PH} = \mathcal{C}$. Does $\mathcal{C} = \mathsf{NP} \cap \mathsf{co\text{-}NP}$? NP? P^{NP}? $\sum_2^p \cap \prod_2^p$? Also conclude that if PH has a complete language under $\leq_T^p$ then it collapses. (18 pts. total)

(5) Define the *Boolean closure* of a complexity class $\mathcal{C}$ is the class of languages that can be written as unions and intersections of languages in $\mathcal{C}$ together with their complements. The class co-$\mathcal{C}$ counts as being in the first level of the closure, and then the minimum number of languages in co-$\mathcal{C} \cup \mathcal{C}$ that are combined is the level.

(a) Show that for any language $A \in \mathsf{NP}$ and $B \in \mathsf{co\text{-}NP}$, their join $A \oplus B$ belongs to the second level of the Boolean closure of NP.

(b) Now consider their splice $E||(A, B)$ by a language $E \in \mathsf{P}$. Show that it also belongs to the Boolean closure of NP. Does it land in the 4th, 3rd, or 2nd level?

(c) Then say why the whole Boolean closure of NP is contained in P^{NP}. ($6 + 6 + 6 = 18$ pts., for 120 total on the set)